

THE OFFICE OF THE STATE CHIEF INFORMATION OFFICER  
ENTERPRISE TECHNOLOGY STRATEGIES

---

North Carolina Statewide Technical Architecture

**Implementation Guidelines:  
Systems Integration Architecture**

STATEWIDE TECHNICAL ARCHITECTURE

# Implementation Guidelines: Systems Integration Architecture

---

Initial Release Date:	May 29, 2003	Version:	1.0.0
Revision Approved Date:	Not Applicable		
Date of Last Review:	March 11, 2004	Version:	1.0.1
Date Retired:			
Architecture Interdependencies:			
Reviewer Notes: Reviewed and updated office title and copyright date. Added a hyperlink for the ETS email – March 11, 2004.			

© 2004 State of North Carolina  
Office of the State Chief Information Officer  
Enterprise Technology Strategies  
PO Box 17209  
Raleigh, North Carolina 27699-7209  
Telephone (919) 981-5510  
[ets@ncmail.net](mailto:ets@ncmail.net)

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any informational storage system without written permission from the copyright owner.

<b>INTRODUCTION .....</b>	<b>3</b>
<b>IMPLEMENTATION GUIDELINES .....</b>	<b>3</b>
IMPLEMENTATION GUIDELINES FOR APPLICATION COMMUNICATION	
MIDDLEWARE.....	3
<i>Guideline 1: Include a design strategy to allow a broker to be easily integrated into the middleware architecture being implemented.....</i>	3
IMPLEMENTATION GUIDELINES FOR APPLICATION COMMUNICATION	
MIDDLEWARE BROKERS.....	4
<i>Guideline 1: Implement decoupled services that are accessible by the service broker as opposed to developing custom interfaces for integrating between applications.....</i>	4
IMPLEMENTATION GUIDELINES FOR APPLICATION INTEGRATION .....	5
<i>Guideline 1: Keep security in mind when designing applications.....</i>	5
<i>Guideline 2: Use screen scraping for static applications that have very little maintenance activity and only when no other interface exists.....</i>	5
<i>Guideline 3: With screen scraping techniques, either the workstation application or the interface engine can be the integration point.....</i>	5
<i>Guideline 4: Screen scraping should be viewed as a short-term (less than two year) solution... </i>	6
<i>Guideline 5: Use screen scraping when the transaction volumes are expected to be low. ....</i>	6
<i>Guideline 6: In program-to-program interfaces, new application systems should invoke existing application logic. This typically requires modification to the existing application in order to accommodate the new application call.....</i>	6
<i>Guideline 7: Application modifications are required for direct program-to-program communications and are recommended when the existing systems are going to continue to be in use for a long period of time.....</i>	7
<i>Guideline 8: Use middleware and application program interfaces (APIs) as the program-to-program interface. The most common use for remote procedures is when existing mainframe application logic is packaged so it can be invoked using a remote procedure call facility. ....</i>	7
<i>Guideline 9: Use an interface engine to link multiple heterogeneous applications, especially when at least one non-invasive link is required.....</i>	7
<i>Guideline 10: The interface engine programming should not contain any application specific business rules; it should only contain the rules required to integrate heterogeneous application systems messages and data.....</i>	8
<i>Guideline 11: When implementing an interface engine, use a non-invasive connection to an existing application whenever possible. ....</i>	8
IMPLEMENTATION GUIDELINES FOR EDI.....	9
<i>Guideline 1: Implement EDI for common business transactions.....</i>	9
<i>Guideline 2: EDI should be implemented using either the X12 or EDIFACT standard. ...</i>	9
<i>Guideline 3: EDI should be implemented using a Value Added Network (VAN). ....</i>	9
IMPLEMENTATION GUIDELINES FOR TERMINAL INTEGRATION .....	10
<i>Guideline 1: Segment applications to separate the user interface on a separate tier.....</i>	10
<i>Guideline 2: Develop new secure applications on the client/server platforms and implement a terminal integration solution to provide SNA 3270 terminal access to the new application systems. ....</i>	10

## Introduction

The intent of this document is to provide general implementation guidelines

with in the Systems Integration Technology Domain. This will help to ensure that the State of North Carolina adopts uniform and consistent implementations of system integration solutions across the enterprise.

The key goal of this document is to outline implementation guidelines that, when followed by the solution developers, will lead to a well-designed solution that has the flexibility to grow with changes in technology and can be maintained in an efficient and effective manner. This is a fundamental principle of the North Carolina Statewide Technical Architecture.

*This implementation guild is currently being revised both to better reflect the state of technology and improve upon existing architectural guidelines to better support state agency initiatives.*

## Implementation Guidelines

### Implementation Guidelines for Application Communication Middleware

Implementation guidelines in this section pertain to types of application communication middleware. Clearly documented application architecture is required prior to selecting a particular inter-application communication middleware solution. In most cases, a combination of application communication middleware will be required.

**Guideline 1: Include a design strategy to allow a broker to be easily integrated into the middleware architecture being implemented.**

#### Rationale

- Brokers simplify coding and maintenance of the middleware.
- Easy integration of a broker will facilitate future inter-application integration.

## Implementation Approach for Application Communication Middleware

Avoid New Deployment Migrate from Technology	Current Technology Direction	Emerging Technology
Monolithic applications where all communication is contained within application	Remote Procedure Call, Message Oriented Middleware, and Distributed Transaction Monitor.	Service Broker or Object Request Broker

*Table 1 - Implementation Approach for Application Communication Middleware*

## Implementation Guidelines for Application Communication Middleware Brokers

When communication external to the application is required, use of a statewide broker is required.

**Guideline 1: Implement decoupled services that are accessible by the service broker as opposed to developing custom interfaces for integrating between applications.**

### Rationale

- The service broker offers a reusable, standard interface that makes shared services available throughout the state.
- It provides the state with a higher level of application development productivity by ensuring a greater degree of consistency, modular development, and simplified testing.

**Implementation Approach for Application Communication Middleware between Applications.**

Avoid New Deployment Migrate from Technology	Current Technology Direction	Emerging Technology
Developing custom interfaces when communicating between application systems	Service Broker	Object Request Broker

*Table 2 - Implementation Approach for Application Communication Middleware between Applications*

## Implementation Guidelines for Application Integration

### **Guideline 1: Keep security in mind when designing applications.**

#### **Rationale**

- Implement a security solution with authentication and access control as needed.
- Know what steps are required to access a remote application system and ensure that users have permission to execute any remote application calls or to log in to a remote system.

### **Guideline 2: Use screen scraping for static applications that have very little maintenance activity and only when no other interface exists.**

#### **Rationale**

- The remote system is unaware of the screen-scraping program. Thus even simple screen modifications to the remote system will adversely affect the operability of a custom screen-scraping program.

### **Guideline 3: With screen scraping techniques, either the workstation application or the interface engine can be the integration point.**

### Rationale

- Rely on IBM's High Level Language Application Program Interface (HLLAPI) interface standards and use a tool that generates standard HLLAPI calls.
- The strength of screen scraping is that no changes are made to the existing application system (e.g., screen scraping is non-invasive).

### **Guideline 4: Screen scraping should be viewed as a short-term (less than two year) solution.**

### Rationale

- Screen scraping is a strategy that is good for bridging the transition from poorly understood or inflexible application systems to applications that are more flexible and better documented.

### **Guideline 5: Use screen scraping when the transaction volumes are expected to be low.**

### Rationale

- Screen scraping programs must process screen-formatting data in addition to application data in order to accomplish integration function. This additional overhead makes screen-scraping programs unsuitable for high volume transactions.

### **Guideline 6: In program-to-program interfaces, new application systems should invoke existing application logic. This typically requires modification to the existing application in order to accommodate the new application call.**

### Rationale

- Application developers should not have to recreate existing business logic within each new application system.
- Using existing business logic has the potential to reduce software development and maintenance costs.

**Guideline 7: Application modifications are required for direct program-to-program communications and are recommended when the existing systems are going to continue to be in use for a long period of time.**

**Rationale**

- Custom programs providing integrative functions can be designed and documented to minimize the complexity of the interface. In the long run, maintenance costs will be lower, especially when the systems being integrated are subject to change.
- Direct program-to-program integration can handle transaction volume more efficiently than screen scraping methods.

**Guideline 8: Use middleware and application program interfaces (APIs) as the program-to-program interface. The most common use for remote procedures is when existing mainframe application logic is packaged so it can be invoked using a remote procedure call facility.**

**Rationale**

- Middleware and APIs provide standard and well-documented means of providing inter-application communication.

**Guideline 9: Use an interface engine to link multiple heterogeneous applications, especially when at least one non-invasive link is required.**

**Rationale**

- An interface engine simplifies the interoperation of heterogeneous applications residing on heterogeneous platforms, having different network protocols, data formats and record layouts. It simplifies the connectivity by imposing only one connection per application instead of many point-to-point connections that would otherwise be necessary.
- If a modification is made to one system, the change may only need to be reflected in the interface engine configuration.



**Guideline 10: The interface engine programming should not contain any application specific business rules; it should only contain the rules required to integrate heterogeneous application systems messages and data.**

**Rationale**

- This minimizes complexity by keeping the business rules in the applications instead of in the interface.

**Guideline 11: When implementing an interface engine, use a non-invasive connection to an existing application whenever possible.**

**Rationale**

- A customized invasive connection creates an additional level of complexity that will be expensive to develop and maintain.

**Implementation Approach for Application Integration**

Avoid New Deployment Migrate From Technology	Current Technology Direction	Emerging Technology
Screen scraping	<p>Use existing application logic.</p> <p>Use middleware and application program interfaces (APIs) as the program-to-program interface.</p> <p>Use an interface engine to link multiple heterogeneous applications, especially when at least one non-invasive link is required.</p>	<p>XML based integration</p> <p>Web Services</p>

*Table 3 - Implementation Approach for Application Integration*

## **Implementation Guidelines for EDI**

### **Guideline 1: Implement EDI for common business transactions.**

#### **Rationale**

- EDI offers industry standard transaction formats and guidelines. EDI facilitates rapid, accurate exchange of standard commercial transactions by creating, transferring and processing the transactions electronically.
- Most large companies are EDI capable for commonly performed transactions.

### **Guideline 2: EDI should be implemented using either the X12 or EDIFACT standard.**

#### **Rationale**

- Most EDI software is capable of generating and interpreting the correct format for each trading partner as long as the format is stored in the trading partner profile.
- The current format for EDI is ANSI X12. EDIFACT is an international standard that is closely coordinated with X12.

### **Guideline 3: EDI should be implemented using a Value Added Network (VAN).**

#### **Rationale**

- Using a VAN relieves the organization from having to schedule transmissions for each trading partner.
- Using a VAN requires the organization to transmit using only a single transmission protocol, that of the VAN, rather than using a different protocol for each trading partner.
- VANs often provide additional services such as reformatting between different EDI standards, forwarding non-EDI documents to trading partners, and splitting transactions into their accounting and non-accounting components and then routing them appropriately.

## Implementation Approach for EDI

Avoid New Deployment Migrate From Technology	Current Technology Direction	Emerging Technology
Avoid performing common business transactions manually	Electronic Data Interchange for common transactions	
Proprietary EDI transaction formats	Standard EDI formats (see Standards below) using X12 or EDIFACT	Extensible Markup Language (XML) to define transactions
Direct EDI	Use a Value Added Network for transaction delivery  Use an Automated Clearing House (ACH) for financial EDI	Use of the Internet for EDI transaction delivery

Table 4 - Implementation Guideline Summary for EDI

## Implementation Guidelines for Terminal Integration

**Guideline 1: Segment applications to separate the user interface on a separate tier.**

## Rationale

- Designing applications whose user interfaces and business rules are incorporated within a single tier does not allow a dumb terminal to integrate into the new system.

**Guideline 2: Develop new secure applications on the client/server platforms and implement a terminal integration solution to provide SNA 3270 terminal access to the new application systems.**

### Rationale

- Systems designed using 3270 terminals and SNA are being phased out gradually and being replaced by modern client/server systems using intelligent workstation technology and the TCP/IP transport protocol. A terminal integration solution that allows 3270 access to client/server systems can extend the functional life of the existing investment of 3270 terminals.

### Implementation Approach Summary for Terminal Integration

Avoid New Deployment Migrate From Technology	Current Technology	Emerging Technology
User interfaces incorporated within single-tier applications.	Segment applications to separate the user interface on a separate tier.	In the future, terminal integration is not required because intelligent workstations will be the prevailing technology.
Developing applications using legacy application architecture because of the large investment in 3270 terminals.	Develop new applications on the client/server platforms and implement a terminal integration solution to provide SNA 3270 terminal access to the new application systems.	

*Table 5 - Implementation Approach Summary for Terminal Integration.*